

15º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2024

ARQUITETURA E DESENVOLVIMENTO DE MICROSERVIÇOS SEGUINDO PADRÕES DE APLICAÇÃO EMPRESARIAL VOLTADOS A EXPLORAÇÃO E MARCAÇÃO GEOGRÁFICAS

H. G. SANTOS¹, A. L. S. CONDE², F. S. SANTOS³

¹ Graduando em Engenharia da Computação, IFSP, Campus Birigui, higor.santos@aluno.ifsp.edu.br

² Graduando em Engenharia da Computação, IFSP, Campus Birigui, andre.conde@aluno.ifsp.edu.br

³ Docente IFSP, Campus Birigui, fsergio.santos@ifsp.edu.br

Área de conhecimento (Tabela CNPq): 1.03.03.04-9 Sistemas de Informação

RESUMO: Este trabalho propõe uma arquitetura de microserviços para a exploração e marcação geográfica de imagens na agricultura de precisão, utilizando inteligência artificial para melhorar a eficiência no uso de recursos. A metodologia envolve a implementação de microserviços em *Python* que operam de forma independente para executar funções específicas no *pipeline* de processamento de imagens. Utilizou-se o programa *QGIS* para a extração de áreas de interesse e a biblioteca *OpenCV* para segmentação e extração de dados georreferenciados. O *Apache Kafka* gerencia a comunicação em tempo real entre os microserviços, garantindo a sincronização e ordenação dos dados processados. A arquitetura proposta facilita a escalabilidade e manutenção, dividindo a aplicação em serviços modulares. A principal limitação foi a capacidade de processamento, mitigada pela subdivisão de imagens em seções menores, o que permitiu um processamento paralelo mais eficiente. Os resultados indicam que, apesar de necessitar ajustes para melhorar a precisão, a solução proposta é promissora para otimizar o manejo de recursos agrícolas, proporcionando uma base sólida para futuras melhorias na área de inteligência artificial aplicada à geografia.

PALAVRAS-CHAVE: Micro serviços; Georreferenciamento; Inteligência Artificial; Agricultura de Precisão; Processamento de Imagens; Automação.

ARCHITECTURE AND DEVELOPMENT OF MICROSERVICES FOLLOWING ENTERPRISE APPLICATION PATTERNS FOR GEOGRAPHIC EXPLORATION AND MARKING.

ABSTRACT: This work proposes a microservices architecture for the exploration and geographic tagging of images in precision agriculture, using artificial intelligence to improve efficiency in the use of resources. The methodology involves implementing microservices in *Python* that operate independently to perform specific functions in the image processing pipeline. The *QGIS* program was used to remove areas of interest and the *OpenCV* library was used to segment and remove georeferenced data. *Apache Kafka* manages real-time communication between microservices, ensuring synchronization and ordering of processed data. The proposed architecture facilitates scalability and maintenance, dividing the application into modular services. The main limitation was the processing capacity, mitigated by subdividing images into smaller packets, which allowed for more efficient parallel processing. The results indicate that, although adjustments are necessary to improve precision, the proposed solution is promising for improving the management of agricultural resources, providing a solid foundation for future improvements in the area of artificial intelligence applied to geography.

KEYWORDS: Micro services; Georeferencing; Artificial intelligence; Precision Agriculture; Image Processing; Automation.

INTRODUÇÃO

A agricultura é uma atividade essencial que engloba o cultivo de plantas e vegetais, como frutas, legumes e verduras, além da criação de animais, como bois, porcos e galinhas, para consumo humano. Com uma grande importância, a agricultura deve ser conduzida com eficiência para maximizar a produtividade e minimizar o desperdício, garantindo assim a sustentabilidade socioeconômica e ambiental. Ao longo dos anos, diversas técnicas e métodos foram desenvolvidos para aumentar a produtividade e reduzir o desperdício.

A agricultura de precisão surge como uma abordagem moderna que visa otimizar o uso de recursos agrícolas. Essa abordagem baseia-se no uso de tecnologias avançadas para otimizar a produção e minimizar os custos e impactos ambientais. Esta técnica permite um manejo mais eficaz dos recursos agrícolas, utilizando tecnologias de georreferenciamento, como o GPS, e técnicas de automação, como drones e maquinaria agrícola automatizada.

A agricultura de precisão, portanto, possibilita um gerenciamento mais eficiente dos recursos, minimizando custos e maximizando a produção. Dentre as tecnologias aplicadas destacam-se o georreferenciamento, o uso de GPS e técnicas de automação, como drones e maquinaria agrícola automatizada.

Atualmente, um dos maiores desafios enfrentados pela agricultura de precisão está na obtenção do georreferenciamento de áreas de interesse em imagens de campos agrícolas. Muitas vezes, esse processo é realizado manualmente através de software como QGIS, uma abordagem que consome tempo significativo e requer mão de obra especializada. A ausência de intervenção humana constante e a possibilidade de erros humanos introduzem incertezas e ineficiências no processo.

Este trabalho apresenta uma solução para a exploração e marcação geográfica de imagens mediante inteligência artificial, com o objetivo de criar uma arquitetura robusta que facilite o processo, melhorando a eficiência da extração de dados georreferenciados. A independência de sistemas e microsserviços é um fator crucial para garantir escalabilidade e eficiência em sistemas complexos como este. A automatização dos processos de obtenção, transformação e validação de coordenadas georreferenciadas é essencial para a modernização da agricultura de precisão, contribuindo para o avanço do conhecimento na área de inteligência artificial aplicada à geografia, bem como para o desenvolvimento de novas ferramentas e aplicações que auxiliem os usuários em suas necessidades geográficas.

MATERIAL E MÉTODOS

Materiais:

- Linguagem de programação de computadores Python versão 3.11;
- Software QGIS versão 3.30.2, aplicação de Sistema de Informações Geográficas gratuito e de código aberto que oferece suporte a visualização, edição e análise de dados geoespaciais;
- OpenCV versão 4.9.0.80, biblioteca de código aberto de software de visão computacional e aprendizado de máquina;
- Acer Nitro 5 - Computador com Placa de Vídeo GTX 1650, 16GB de memória RAM e processador CPU AMD Ryzen 7 4800H, 8 núcleos com Radeon Graphics, 2,90 GHz;
- Kafka 3.4, utilizado como principal serviço de mensageria;
- Zookeeper 3.9.2, fornece os serviços de *cluster* distribuído para o Kafka;
- Confluent Kafka 2.3.0, biblioteca Python para comunicação com Kafka;
- FastAPI 0.110.0, permite a criação de API's HTTP.

Métodos:

O projeto foi implementado em Python, uma linguagem amplamente utilizada para tarefas de processamento de dados e automação. Cada microsserviço foi desenvolvido para executar uma função específica dentro do *pipeline* de extração, e a integração com Apache Kafka foi essencial para a comunicação e gerenciamento dos fluxos de dados em tempo real.

Kafka é uma plataforma de *streaming* distribuída que permite a publicação, armazenamento e processamento de fluxos de dados em tempo real. Serviços de mensageria são amplamente utilizados para a construção de *pipelines* de dados de alta performance e sistemas de integração em tempo real,

pois sua arquitetura robusta permite a manipulação de grandes volumes de dados, com alta confiabilidade e baixa latência. No contexto deste projeto, o *Kafka* é utilizado para orquestrar a comunicação entre os micros serviços a partir de um serviço principal, garantindo que todos os dados estejam sincronizados e que as operações ocorram de forma sequencial e ordenada.

Um software composto por microsserviços divide uma aplicação em vários serviços pequenos, cada um responsável por uma parte específica da funcionalidade. Esses serviços são independentes e se comunicam entre si, geralmente através de APIs ou mensagens. A principal vantagem dessa abordagem é a escalabilidade e a possibilidade de manutenção independente de cada componente. Utilizou-se de uma arquitetura baseada em micro serviços, a fim de garantir um princípio de responsabilidade única, a TABELA 1 descreve a divisão das etapas do processamento em serviços.

TABELA 1. Etapas e microsserviços

Etapa	Microsserviços	Descrição
Extraír a área de interesse	<i>Dionísio</i>	A partir de uma <i>bounding box</i> , recorta o talhão de interesse.
Recorte em subseções	<i>Ares</i>	Divide a área de interesse em secções com dimensão máxima 4096 pixels.
Segmentação	<i>Laquesís</i>	Utiliza uma rede neural para segmentar a imagem.
Extração de esqueleto	<i>Hermes</i>	Extraí o esqueleto do campo utilizando os processos de erosão e dilatação na saída segmentas do serviço <i>Laquesís</i> , obtendo o esqueleto representados por pixels.
Sintetizar os dados obtidos nas secções em uma única secção.	<i>Demeter</i>	A partir da extração do esqueleto, realiza a sintetização dos dados para um único campo.
Converte os dados de pixel para georreferência	<i>Apolo</i>	Realiza a conversão dos dados de pixel, para georreferência

O serviço principal foi nomeado *Hera*, dedicado a gerenciar o acesso *HTTP*, supervisionar e atualizar o banco de dados, uma coleção organizada de informações que podem ser facilmente acessadas, gerenciadas e atualizadas. Neste projeto utilizou-se o PostgreSQL, um gerenciador de banco de dados relacional. Os dados processados pelas várias etapas do pipeline são armazenados em um banco de dados para posterior análise e consulta. A persistência desses dados é crucial para garantir que as informações estejam sempre disponíveis e que possam ser utilizadas para gerar *insights* e otimizar processos futuros.

A FIGURA 1 retrata um esquemático que representa a comunicação entre os microsserviços e o serviço principal.

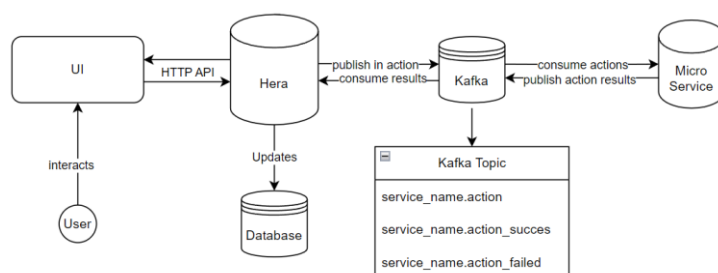


FIGURA 1: Fluxo e dados. FONTE: Elaborado pelo autor.

Este serviço também é responsável por manejar os tópicos do *Kafka*, controlando os eventos de domínio da aplicação. Ele atua como um ponto central de integração, garantindo que as comunicações entre o sistema e os clientes sejam eficientes e seguras, enquanto também orquestra a sincronização dos dados armazenados com os eventos transmitidos pelo *Kafka*. O diagrama esquemático (FIGURA 2) ilustra de forma simplificada a comunicação entre os demais serviços da aplicação.

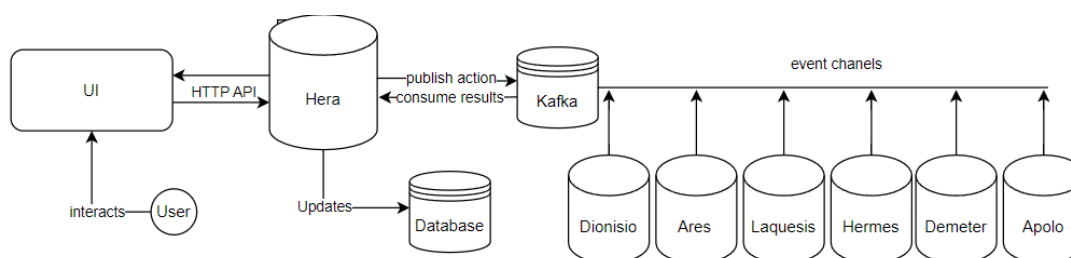


FIGURA 2: Fluxo e dados entre os serviços da aplicação. FONTE: Elaborado pelo autor.

Cada microsserviço dispõe de um tópico para notificar os dados de sucesso e um para falha, onde são publicadas as respectivas causas, facilitando a atualização contínua do banco de dados com as informações mais recentes, assegurando que o sistema opere de forma coesa e responsiva em tempo real. A TABELA 2 está apresentando os tópicos criados para comunicação entre os serviços.

TABELA 2. Tópicos Kafka para transição de dados entre os serviços.

Serviço	Canal	Serviço	Canal
Dionísio	<i>dionisio.box-to-plot</i>	Hermes	<i>hermes.extract-skeleton</i>
	<i>dionisio.created-plot</i>		<i>hermes.skeleton-stracted</i>
	<i>dionisio.extract-failed</i>		<i>hermes.extract-failed</i>
Ares	<i>ares.plot-to-crops</i>	Demeter	<i>demeter.crops-to-plot</i>
	<i>ares.crop-created</i>		<i>demeter.crops-ploted</i>
	<i>ares.crop-failed</i>		<i>demeter.plot-failed</i>
Laquesis	<i>laquesis.segment-crop</i>	Apolo	<i>apolo.pixel-to-geojson</i>
	<i>laquesis.segmented-crop</i>		<i>apolo.conversed-crop</i>
	<i>laquesis.segmentation-failed</i>		<i>apolo.conversion-failed</i>

As imagens utilizadas no processamento foram extraídas de um software *GIS*, chamado *QGIS*, utilizando uma *bounding box* – um conjunto de pontos, que define um retângulo, a área de interesse – aplicada a camada do Google Satélite. Cada imagem extraída continha aproximadamente uma dimensão de 25.000 pixels.

Uma das principais limitações identificadas no projeto foi a capacidade de processamento limitada, decorrente do grande tamanho das imagens utilizadas. O elevado volume de dados presente em cada imagem gerou desafios consideráveis para um processamento eficiente e ágil. O alto número de pixels em cada imagem resultou em uma carga computacional substancial, demandando recursos adicionais para a extração e análise dos dados georreferenciados. No caso específico, o equipamento utilizado foi um *Acer Nitro 5*. A principal limitação observada foi a memória *RAM*, que não foi suficiente para lidar de forma eficiente com o grande volume de dados gerados pelas imagens de alta resolução. Durante o processamento, frequentemente houve lentidão e dificuldades para manter o sistema operacional e os aplicativos funcionando de forma fluida devido à sobrecarga de memória.

Para mitigar essa limitação, as imagens foram subdivididas em seções com dimensão máxima de 4096 pixels, com resoluções reduzidas, que foram processadas de forma independente. Após a conclusão do processamento dessas seções menores, os resultados foram reunidos e integrados em um resultado único, correspondente a imagem consolidada. Essa abordagem não apenas facilitou o manejo dos dados, como também melhorou a eficiência do sistema, permitindo que o sistema como um todo, processe múltiplas etapas de uma mesma imagem e múltiplas imagens em uma mesma etapa paralelamente.



FIGURA 3. Área de interesse extraída do software *QGIS* com talhão de interesse destacada em vermelho. Fonte: Elaborada pelo autor.

RESULTADOS E DISCUSSÃO

As etapas de recorte, segmentação e extração de esqueletos podem ser visualizadas a partir da FIGURA 4.

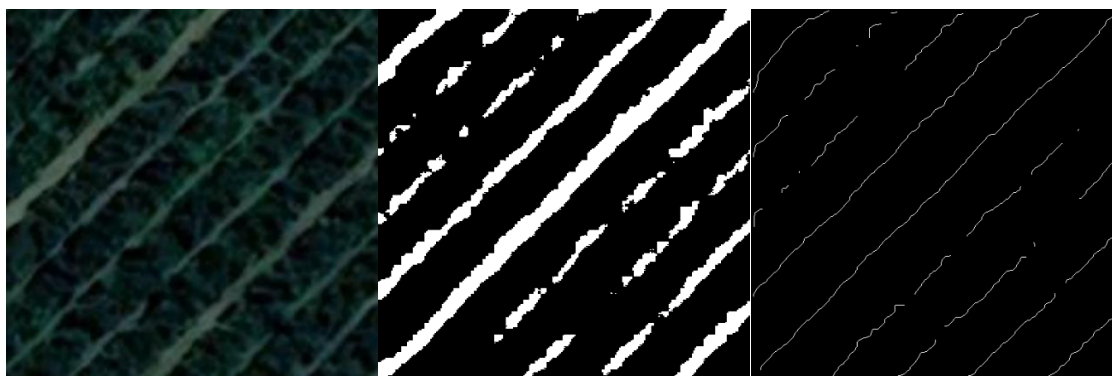


FIGURA 4. Secção da área de interesse, segmentação e extração do esqueleto da área de interesse. Fonte: Elaborada pelo autor.

Após a obtenção do esqueleto da imagem, foi aplicada a Transformada de *Hough* (TH), disponível na biblioteca *OpenCV*, para identificar os pontos inicial e final de cada linha na imagem. Com os pontos de início e fim de cada linha, juntamente com as posições georreferenciadas do ponto superior esquerdo e inferior direito correspondentes as posições dos pixels foram convertidas para coordenadas geográficas.

A escolha e ajuste dos parâmetros da TH foram realizados de forma iterativa, visando otimizar a identificação das linhas de interesse, levando em conta a complexidade das estruturas presentes nas imagens e a precisão desejada para as coordenadas georreferenciadas. Esse processo exigiu ajustes iterativos dos parâmetros da TH, buscando um equilíbrio entre sensibilidade e precisão para captar adequadamente linhas com orientações variadas e diferentes níveis de detalhe. A TH foi executada em múltiplas escalas, permitindo a detecção robusta em imagens com variações estruturais significativas, como mudanças na espessura e na continuidade das linhas.

Além disso, foram realizados testes com diferentes resoluções de detecção e limiares para reduzir a incidência de linhas redundantes ou descontinuidades indesejadas, melhorando a clareza dos resultados. Após a identificação das linhas, utilizamos pontos de referência georreferenciados, como os cantos superior esquerdo e inferior direito da imagem, para converter as coordenadas dos pixels em coordenadas geográficas. Isso possibilitou a representação espacial precisa das informações extraídas, facilitando sua aplicação prática em análises espaciais e sistemas de informação geográfica (SIG).

Como Resultado do processamento obtém-se um *GeoJson* com as coordenadas espaciais obtidas. Importando para o *QGIS* foi possível visualizar as linhas detectadas (FIGURA 5).

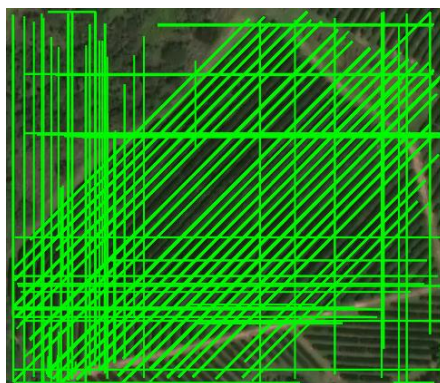


FIGURA 5: Captura de tela do software QGIS contendo o arquivo GeoJSON gerado. Fonte: Elaborado pelo autor.

Este trabalho foca na obtenção da posição geográfica exata das linhas de carreadores, um objetivo que requer precisão adicional na análise e no processamento dos dados. Sendo assim, agrega um valor significativo, pois permite a representação das linhas em um contexto geográfico, facilitando sua visualização.

CONCLUSÕES

O presente trabalho conseguiu atingir parcialmente o objetivo proposto, que era obter os dados georreferenciados dos carreadores. No entanto, o uso da Transformada de *Hough* para a identificação das linhas não produziu os resultados esperados, apresentando limitações. Houve falhas no processamento de curvas de nível, e o algoritmo detectou múltiplas linhas onde deveria haver apenas uma, comprometendo a clareza e precisão dos resultados.

A exploração dos parâmetros da Transformada de *Hough* mostrou que ajustes específicos melhoram a detecção, mas não foram suficientes para resolver todos os problemas observados. Esse cenário sugere a necessidade de aprimoramentos metodológicos. Entre eles, destaca-se a possibilidade de ajustar modelos matemáticos, integrar algoritmos complementares, ou refinar o pré-processamento dos dados para mitigar essas limitações e obter melhores resultados.

CONTRIBUIÇÕES DOS AUTORES

H.G.S. e A.L.S.C. contribuíram com a metodologia e testes. A.L.S.C. foi responsável pela análise dos dados, criação da rede neural e desenvolvimento dos algoritmos de Processamento de Imagens. H.G.S. atuou no desenvolvimento dos algoritmos de Processamento de Imagens e desenvolvimento dos micros serviços para aplicação dos algoritmos desenvolvidos nas imagens. F.S.S. atuou na redação do trabalho e forneceu orientações para o desenvolvimento da metodologia.

Todos os autores contribuíram com a revisão do trabalho e aprovaram a versão submetida.

AGRADECIMENTOS

Agradecemos ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) pelo suporte institucional e ao orientador pelo auxílio para o desenvolvimento deste trabalho.

REFERÊNCIAS

- APACHE KAFKA. *Apache Kafka Documentation*. Disponível em: <https://kafka.apache.org/>. Acesso em: 1 set. 2024.
- FLOWER, M., RICE, D., FOEMEEL, M., HIEATT, E., MEE, R., & STAFFORD, R. (2003). *et al. PATTERNS OF ENTERPRISE APPLICATION ARCHITECTURE*. Addison-Wesley.
- KESTUR, R. (2016). **Detection of Rows in Agricultural Crop Images Acquired by Remote Sensing from a UAV**. International Journal of Image, Graphics and Signal Processing. 8. 25-31. 10.5815/ijigsp.2016.11.04.
- LAMAS, F. M. **A evolução da agricultura do brasil**. EMBRAPA, 2023. Acesso em 24 jun. 2024 Disponível em: <https://www.embrapa.br/busca-de-noticias/-/noticia/81665485/artigo---a-evolucao-da-agricultura-do-brasil>

RONNEBERGER, O.; FISCHER, P.; BROX, T. (2015). **U-Net: Convolutional Networks for Biomedical Image Segmentation**. <https://doi.org/10.48550/ARXIV.1505.04597>

SOARES, G. A. **Identificação de linhas de plantio por meio da Transformada de Hough**. 2017. 47 f. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) - Universidade Federal de Uberlândia, Uberlândia, 2017.

TAN, M.; LE, Q. V. (2019). **EfficientNet: Rethinking model scaling for convolutional Neural Networks**. <https://doi.org/10.48550/ARXIV.1905.11946>